

### **REMARKS**

Upon entry of the present amendment, claims 14-20 are canceled, claim 10 is amended and claim 21 is added, leaving claims 10-13 and 21 pending with claim 10 being independent.

Claims 10-14, 17 and 18 stand rejected under 35 USC §103(a) as being unpatentable over US Patent No. 5,828,884 to Lee et al. in view of US Patent No. 6,349,344 to Sauntry. The Examiner contends that the combination of these references renders the above claims obvious. Applicants respectfully traverse this rejection.

The present claims are drawn to a circuit arrangement for use in a microprocessor which allows for automatic type conversion based on a data type stored together with the data to be processed. This makes it possible to use the same machine code for processing data of different types because the machine code needs not be aware of the type of data stored in the processor registers involved.

Sauntry merely deals with software (multiple Java classes combined in a run-time image which comprises preloaded and prepared class files suitable for interpretation by a Java virtual machine, see abstract), not hardware. The present application is directed at hardware for execution of software.

Data type conversion is a common task performed by software which generally runs on microprocessors; however, the fact that software exists that performs data type conversion does not in any way imply a microprocessor that stores an indication of the type of data stored in a processor register together with the data itself and evaluates this indication during execution of processor instructions in order to allow for machine code that can process different types of data without any change to the machine code. This is clear considering the fact that software performing type conversion is presently executed on processors not being distinguished by the new feature proposed by the present application and which require execution of additional processor instructions and multiple instances of the same code each being adapted to treat a specific type of data which is precisely the shortcoming of the prior art which is addressed by the present invention.

Lee deals with the problem that software being executed by a processor should not have to be concerned about the endianness of commanding data sent to other hardware instances which may have varying endiannesses. If the endianness of data is understood as a data type, Lee

can be interpreted as allowing for code which can process data independent of their data type. Because according to Lee, the code being executed needs not be aware of the endianness of the data being handled.

However, the solution disclosed in Lee significantly differs from that of the present application. According to Lee, the processor will declare some ranges of memory addresses (conversion apertures, see col. 6, lines 34 through 39) to a byte swapping device which is situated between the processor and the storage device. The byte swapping device will monitor the addresses on the address bus and will perform a byte swapping operation while passing on data between the processor and the storage device if the address on the address bus falls within the range of one of the conversion apertures and pass on the unchanged data else.

Therefore, Lee proposes to declare ranges of data as belonging to a specific data type prior to execution of data type independent code. When this data is being processed, conversion will be performed.

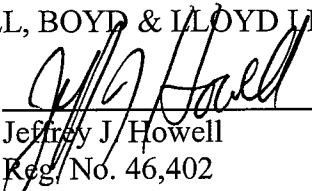
Applicants submit that claim 10 and its dependent claims are allowable over the cited prior art. Moreover, new dependent claim 21 further distinguished itself from the prior art.

The Commissioner is hereby authorized to charge deposit account 02-1818 for any fees which are due and owing.

Respectfully submitted,

BELL, BOYD & LLOYD LLP

BY

  
\_\_\_\_\_  
Jeffrey J. Howell  
Reg. No. 46,402  
Customer No. 29180

Dated: October 9, 2007